UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/825,249	04/02/2001	Rahul Sharma	SUNMP006	2037

25920   7590   07/14/2006

MARTINE PENILLA & GENCARELLA, LLP
710 LAKEWAY DRIVE
SUITE 200
SUNNYVALE, CA 94085

EXAMINER
PHAM, CHRYSTINE

ART UNIT	PAPER NUMBER
2192	

DATE MAILED: 07/14/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

	Application No.	Applicant(s)
Office Action Summary	09/825,249	SHARMA ET AL.
	Examiner	Art Unit	
	Chrystine Pham	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *20 April 2006*.

2a)☒ This action is **FINAL**.   2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,4-9,13-21 and 25* is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1, 4-9, 13-21, and 25* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☐ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
    Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____.

5)☐ Notice of Informal Patent Application (PTO-152)

6)☐ Other: _____.

DETAILED ACTION

1.   This action is responsive to the Amendment filed on April 20, 2006. Claims 1, 4-

5, 9, 13, 18, and 25 have been amended. Claims 2-3, 10-12, 22-24 have been

canceled. Claims 1, 4-9, 13-21, and 25 are pending.

Response to Arguments

2.   Applicant's arguments filed April 20, 2006 have been fully considered but they

are not persuasive.

       With respect to Applicants' remark that the "levels of importance" (related

to the claimed replicated state management types) need not appear, in the

claims (Remarks, starting on page 8, last paragraph), since "levels of

importance" do not appear in the claim, Applicants' remark is deemed irrelevant

to the invention, as claimed.

       Applicants essentially contend, "Apte does not teach recoverable state

being one of memory type and disk type" (Remarks, page 15, under **subsection

2.2.A**). The Examiner respectfully disagrees.

       Col.15:8-18 of Apte explicitly discloses that it is necessary for some server

(EJB) beans to be **temporarily displaced from memory**, rather than destroy the

server bean and lose its ability to continue functioning on behalf of the client, the

state information (i.e., state object) of the server bean may be stored so that the

server bean can be resurrected. This feature of beans is called "persistence". It

is clear from this passage that displacing the server beans from memory and

storing its state object so that it can be subsequently resurrected, i.e., bean

persistence, clearly anticipates a disk replicated state management type since if

a bean is not stored in memory (i.e., memory replicated state management type),

it has to persisted (i.e., disk replicated state management type) for resurrection.

Applicants essentially contend, "Continuing with the review of FIG.11, at

C15, L 14+, reference is made to storing the state information of the server bean,

which is not a state object based on the EJB (client object)" (Remarks, page 17,

first full paragraph). Applicants appear to suggest that the server bean is not an

EJB. The Examiner respectfully disagrees.

As has been established in the previous Office Action (page 5), FIG.12 of

Apte explicitly discloses the server beans as EJBs 1208 and 1220. Furthermore,

in col.7:30-50, Apte explicitly discloses server beans as EJBs which are

container persisted (i.e., memory replicated state management type) by having

their state saved within the container (i.e., memory). Col.17:40-col.18:28

(associated with FIG.12), Apte explicitly discloses that when an EJB that refers to

another EJB needs to persist, the Tie object performs the stringify operations for

storing the **referred EJB** in the **back-end data store**. The same passage further

discloses the Tie object 1206 retrieving the persisted object reference from

backend storage 1214 to reconstruct object reference 1208. Thus, contrary to

Applicants' argument, Apte's EJB container and back-end storage clearly

anticipate two different types (i.e., memory and disk) replicated state

management.

Applicants essentially contend, "Apte does not teach a state server for a

particular state object being dedicated to a particular state management type"

(Remarks, page 19, 2nd full paragraph).

As discussed above, the server 1202 is dedicated to container-managed

EJBs 1208 (i.e., memory replicated state management) while back-end storage

1222 is dedicated to persisting referred EJBs for future resurrections (i.e., risk

replicated state management).

Applicants further contend, "the cited server 1202 shown in FIG.12 is not

the claimed 'Java server process', but instead is the CORBA-compliant server"

(Remarks, page 18, last paragraph). Applicants appear to suggest that a

CORBA-compliant server is somehow not capable of running EJBs. The

Examiner respectfully disagrees.

Again, FIG.12 of Apte clearly discloses server 1202 comprising EJBs

1208. In col.12:38-42, Apte explicitly discloses that the Java client invokes

remote business methods from an EJB running in a CORBA server. Needless to

say, each EJB that is running on the server clearly anticipates a Java server

process.

3.   In view of the fore going discussion, rejection of claims under USC 102(e) and

103(a) is considered proper and maintained.

Claim Rejections - 35 USC § 102

4.   The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that

form the basis for the rejections under this section made in this Office action:

> *A person shall be entitled to a patent unless –*
>
> *(e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.*

5.   Claims 1, 4, and 5 are rejected under 35 U.S.C. 102(e) as being anticipated by

Apte et al. of record (US 6269373 B1, *Apte et al.*).

        As per claim 1, *Apte et al.* teach a method and system for managing

container-managed state for a Java base application, comprising the operations

of:

- o   classifying individual entity bean objects (see at least *EJB 1208* FIG.12

        & associated text) with a particular modular state management type

        (see at least *1204 Container* FIG.12 & associated text), the state

        management type being one of a recoverable state or a non-

        recoverable state, the recoverable state being one of a memory

replicated state management type and a disk replicated state

management type (see at least *1222 Back-end storage* FIG.12 &

associated text;);

o   providing a plurality of modular state objects/state partitions, each state

object storing a state of a corresponding entity bean object (see at

least *1208 EJB, 1210-1214, 1206 Tie* FIG.12 & associated text) within

a memory address space of a Java server process (see at least *1202

server* FIG.12 & associated text), wherein each state object is

associated with the state management type of the corresponding entity

bean object (see at least *1208, 1206, 1204* FIG.12 & associated text);

and

o   providing state management (see at least *1204 Container* Fig.12 &

associated text) for each entity bean object (see at least *1208 EJB*

Fig.12 & associated text) using a state object associated with the state

management type corresponding to the respective entity bean object

(see at least *1202 Server, 1204 Container, 1206 Tie, 1208 EJB* Fig.12

& associated text; *application's state information, container, persist

references, other servers* col.1:40-col.2:5), the providing state

management being based separately on each different state

management type and on those state objects corresponding to the

different state management type (see at least *1206 Tie, 1204

Container, 1208 EJB* FIG.12 & associated text) the providing state

management comprising replicating each one of the plurality of state

objects is replicated in a state server, a different one of the state

servers being dedicated to a particular one of the state management

types, a different one of the state servers being provided for each

different recoverable state management type (see at least *1208, 1206,*

1204 FIG.12 & associated text; *application's state information,*

container, persist references, other servers col.1:40-col.2:5).

As per claim 4, the rejection of base claim 1 is incorporated. *Apte et al.*

further teach the operation of grouping the state objects based on the type of

state management to which the corresponding entity bean object is classified

(see at least *EJBs, container, protocol, persistence* col.7:25-55).

As per claim 5, *Apte et al.* teach the method as applied to claim 4, wherein

the state management type (see at least *1204* FIG.12 & associated text) into

which a group of state objects are grouped (see at least *1210-1214* FIG.12 &

associated text) identifies a policy for replication of the group of state objects to

the dedicated state server (see at least *1202* FIG.12 & associated text) that is

dedicated to the particular state management type corresponding to the group

(see at least *EJBs, protocol, particular server, mechanisms, persistence,*

container col.7:25-55).

Claim Rejections - 35 USC § 103

6.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically
> disclosed or described as set forth in section 102 of this title, if the
> differences between the subject matter sought to be patented and the
> prior art are such that the subject matter as a whole would have been
> obvious at the time the invention was made to a person having ordinary
> skill in the art to which said subject matter pertains. Patentability shall not
> be negatived by the manner in which the invention was made.

7.   Claims 6-7, 9, 13-20, and 25 are rejected under 35 U.S.C. 103(a) as being

unpatentable over *Apte et al.* in view of Chung et al. of record (US

6105148,*Chung et al.*).

As per claim 6, *Apte et al.* teach the method as applied in to claim 4. *Apte*

et al. do not expressly disclose the state management type identifying a policy for

migration of a state object from one server process to another server process.

However, *Chung et al.* teach the a method and system for providing different

types of state management (e.g., see *volatile state 30 & persistent state 120*

FIG.1 & associated text, col.2:6-11, col.5:53-60) for entity bean objects (e.g.,

FIG.8A & associated text) wherein checkpoints are managed using state objects

(e.g., FIG.4 & associated text, col.2:62-66, col.4:50-55, col.8:1-3) and state

management unit identifies a particular mechanism for recovery of states for

entity bean objects (e.g., col.2:62-66, col.4:50-55), which are migration capable

between server processes (e.g., FIG.2 & associated text, col.5:10-13). It would

have been obvious to one of ordinary skill in the pertinent art at the time the

invention was made to incorporate the teaching of *Chung et al.* into that of *Apte*

et al. which would produce the expected result with reasonable success. And the

motivation for combining the teachings would have been that utilizing state

objects in managing checkpoints enables the monitoring and persisting of the

states as well as detection of data conflicts which might occur following each

checkpointed state, thus, enforcing data consistency and allowing the recovery

(based on the methods specified in the recovery mechanism) of the application

process to it previous state. Furthermore, it would have been obvious to one of

ordinary skill in the pertinent art at the time the invention was made that

specifying migration mechanism using state objects within state management

units enables the application in the first processing server to be exported to,

installed, and deployed on a second processing server in the event of permanent

or long-term hardware failure of the first server (see at least *Chung et al.* col.5:5-

13).

As per claim 7, the rejection of base claim 1 is incorporated. Claim recites

limitations, which have been addressed in claim 6, therefore, is rejected for the

same reasons as cited in claim 6.

As per claim 9, *Apte et al.* teach a method for managing container-

managed state for a Java application, comprising the operations of:

Partitioning individual entity bean objects of the Java application into state

partitions, wherein the state partitions manage concurrency for the Java

application (see at least *EJBs, mechanisms, concurrency, behavior, container*

col.7:25-55), the partitioning being by storing state of each particular entity bean

object in a state object dedicated to a state management type corresponding to

the state management type of the particular entity bean object (see at least *1208,*

1210-1214, 1206, 1204 FIG.12 & associated text);

Classifying individual state objects within each state partition using state

management units, wherein each particular state management unit is a collection

of the state objects corresponding to one particular state management type for

recoverable state of the respective corresponding particular entity bean objects

(see at least *1208, 1210-1214, 1206, 1204* FIG.12 & associated text); and

Replicating each particular state management unit in one of a plurality of

state servers (see at least *server 104* FIG.1 & associated text; *additional servers*

col.3:58-col.4:5) according to the particular state management type that

corresponds to the particular state objects classified in the particular state

management unit (see at least *EJBs, protocol, particular server, mechanisms,*

persistence, container col.7:25-55).

Apte et al. do not expressly disclose migration capable state of the respective

corresponding entity bean objects. However, *Chung et al.* disclose managing migration

capable state of the processes (see claim 6). It would have been obvious to one of

ordinary skill in the pertinent art at the time the invention was made to incorporate the

teaching of *Chung et al.* into that of *Apte et al.* for the inclusion of migration capable

state. And the motivation for doing so would have been the same as has been cited in

claim 6.

As per claim 13, the rejection of base claim 9 is incorporated. *Apte et al.* further

teach the operation of using a control module/repository (maintaining state

partition specifications) to manage dynamic partitioning/replication of the state of

the application via the state partitions and the state management units (see at

least *container, mechanisms, concurrency* col.7:25-55).

As per claim 14, the rejection of base claim 13 is incorporated. *Apte et al.*

further teach wherein the state partitions and state management units are

modular (see at least *1208 EJB, 1210-1214, 1206 Tie, 1204 Container* FIG.12 &

associated text).

As per claim 15, the rejection of base claim 14 is incorporated. *Apte et al.*

further teach wherein additional state management types for the state

management units can be defined (see at least *1208 EJB, 1210-1214, 1206 Tie,*

1204 Container, 1202 Server FIG.12 & associated text; col.3:58-col.4:5; *EJBs,*

server, protocol col.7:25-55).

As per claim 16, *Apte et al.* teach the method as applied to claim 15. *Apte et al.* further disclose a method and system wherein each state partition serialize transactions for entity bean objects within a particular state partition (e.g., col.15:21-27, col.15:67-col.16:5, col.16:57-65). *Apte et al.* further disclose entity bean objects (e.g., see *EJB 1208* FIG.12 & associated text) of the application are partitioned into state partitions during pre-deployment (e.g., see Abstract, *fields 1210-1214* FIG.12 & associated text).

As per claim 17, the rejection of base claim 16 is incorporated. *Apte et al.* further teach each state partition allows only one concurrent transaction to be performed on the entity bean objects within the particular state partition during a given time period (see at least *container, mechanisms, concurrency* col.7:25-55).

As per claim 18, *Apte et al.* teach a system application for managing managed container-managed state for a Java based application (see at least *704, 728* FIG.7 & associated text), comprising:

o   an application having a plurality of entity bean objects (see at least *1208* Fig.12 & associated text), each entity bean object comprising a state management type (see at least *1204* FIG.12 & associated text), the state management type being one of a recoverable state or non-recoverable state (see at least *1222* FIG.12 & associated text), the recoverable state being one of a memory replicated state management

type and a disk replicated state management type (see at least *1108,*

1112 FIG.11 & associated text);

o   a plurality of state objects (see at least *1210-1214* FIG.12 & associated

text), each state object storing a state of a corresponding entity bean

object (see at least *1208* FIG.12 & associated text) within a memory

address space of a Java server process (see at least *1202* FIG.12 &

associated text), wherein each state object is associated with a

particular state management type of the corresponding entity bean

object (see at least *1210-1214, 1208, 1206, 1204, 1202* FIG.12 &

associated text); and

o   a plurality of state management units (see at least *additional servers*

col.3:55-col.4:5) that classify the state objects, a particular state object

being classified into a particular state management unit based on the

particular state management type of the corresponding entity bean

object wherein the state management units facilitate state

management for each entity bean object;

o   a state server dedicated to each state management type, the state

management type identifying a policy for replication of a state object to

a state server dedicated to a particular state management type (see at

least *EJBs, protocol, particular server, mechanisms, persistence,*

container col.7:25-55); and

  o a replicated state manager configured to replicate a particular state

management unit to the state server that is dedicated to the particular

state management type of the particular state object that is classified

into the particular state management unit to be replicated (see at least

EJBs, protocol, particular server, mechanisms, persistence, container

col.7:25-55).

Apte et al. do not expressly disclose a policy for migration of a state object from one

server process to another server process. However, *Chung et al.* disclose a policy for

migration of a state object from one server process to another server process (see claim

6). It would have been obvious to one of ordinary skill in the pertinent art at the time the

invention was made to incorporate the teaching of *Chung et al.* into that of *Apte et al.* for

the inclusion of a policy for migration of a state object from one server process to

another server process. And the motivation for doing so would have been the same as

has been cited for claim 6.

   As per claim 19, see claim 16.

   As per claim 20, see claim 13.

   As per claim 25, see claim 6.

8.   Claim 8 is rejected under 35 U.S.C. 103(a) as being unpatentable over *Apte et al.*

in view of Nally et al. of record (US 6298478 B1, *Nally et al.*).

        As per claim 8, the rejection of base claim 1 is incorporated. *Apte et al.* do

not expressly disclose the operation of performing lock management using the

state objects. However, *Nally et al.* teach the operation of performing lock

management using the state objects (e.g., *transaction isolation, instances, EJB*

col.3:49-col.4:43). *Apte et al.* and *Nally et al.* are analogous art because they are

both directed to persisting state information for EJBs. It would have been

obvious to one of ordinary skill in the pertinent art at the time the invention was

made to incorporate the teaching of *Nally et al.* into that of *Apte et al.* for the

inclusion of performing lock management using the state objects. And the

motivation for doing so would have been to avoid the performance penalties

inherent in the conventional lock management using serialization, thus enables

multiple concurrent transactions/accesses to the same entity bean object while

ensuring consistency and independent views among the different transactions

(see at least *Nally et al.* col.3:45-col.4:45).

9.   Claim 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over *Apte et*

al. in view of *Chung et al.* further in view of Savage et al. of record (US 6604110,

Savage et al.).

As per claim 21, *Apte et al.* teach the system as applied to claim 20

wherein the repository manages replication of the state of the Java application

during runtime (e.g., see claim 13). *Apte et al.* do not expressly disclose the

repository manages migration of state of the Java application. However, *Savage*

et al. disclose a repository (e.g., see *generic metadata repository 200* FIG.13 &

associated text) managing migration of enterprise application data (e.g., see

generate migration specifications 202 FIG.13 & associated text, col.1:22-25 & 52-

56, col.21:1-6). It would have been obvious to one of ordinary skill in the

pertinent art at the time the invention was made to incorporate the teaching of

Savage et al. into that of *Apte et al.* which would produce the expected result with

reasonable success. And the motivation for combining the teachings would have

been that a repository, which specifies migration protocol, enables the source

application data (e.g., properties, fields, states) persisted in the repository of one

operational system to be analyzed in order to generate metadata/migration

protocol which would specify how the data on that particular operational system

are logically transformed (or made independent) from the underlying operational

system model to other logical and physical structure of data warehouses

(aligning with target business/enterprise structures) on other operational systems

so that said data can be logically mapped, cross-referenced, or incorporated into

diverse type business/enterprise applications.

Conclusion

10.   Applicant's amendment necessitated the new ground(s) of rejection presented in

this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP

§ 706.07(a). Applicant is reminded of the extension of time policy as set forth in

37 CFR 1.136(a).

    A shortened statutory period for reply to this final action is set to expire

THREE MONTHS from the mailing date of this action. In the event a first reply is

filed within TWO MONTHS of the mailing date of this final action and the advisory

action is not mailed until after the end of the THREE-MONTH shortened statutory

period, then the shortened statutory period will expire on the date the advisory

action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be

calculated from the mailing date of the advisory action. In no event, however, will

the statutory period for reply expire later than SIX MONTHS from the date of this

final action.

11.   Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Chrystine Pham whose telephone number is 571-

272-3702. The examiner can normally be reached on Mon-Fri, 8:30am-5pm.

    If attempts to reach the examiner by telephone are unsuccessful, the

examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax

phone number for the organization where this application or proceeding is

assigned is 703-872-9306.

Information regarding the status of an application may be obtained from

the Patent Application Information Retrieval (PAIR) system. Status information

for published applications may be obtained from either Private PAIR or Public

PAIR. Status information for unpublished applications is available through

Private PAIR only. For more information about the PAIR system, see http://pair-

direct.uspto.gov. Should you have questions on access to the Private PAIR

system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-

free).

CP
January 9, 2006

TUAN DAM
SUPERVISORY PATENT EXAMINER